UNO is hard, even for a single player

Erik D. Demaine[1], Martin L. Demaine[1], Ryuhei Uehara[2],
Takeaki Uno[3], and Yushi Uno[4]

[1] MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA. edemaine@mit.edu, mdemaine@mit.edu
[2] School of Information Science, JAIST, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. uehara@jaist.ac.jp
[3] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. uno@nii.jp
[4] Graduate School of Science, Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai 599-8531, Japan. uno@mi.s.osakafu-u.ac.jp

Abstract. UNO® is one of the world-wide well-known and popular card games. We investigate UNO from the viewpoint of combinatorial algorithmic game theory by giving some simple and concise mathematical models for it. They include cooperative and uncooperative versions of UNO, for example. As a result of analyzing their computational complexities, we prove that even a single-player version of UNO is NP-complete, while it becomes in P in some restricted cases. We also show that uncooperative two-player's version is PSPACE-complete.

1 Introduction

Playing games and puzzles is a lot of fun for everybody, and analyzing games and puzzles has long been attracted much interests of both mathematicians and computer scientists [5, 8]. Among various interests and directions of researchers in mathematics and computer science, one of the central issues is their computational complexities, that is, how hard or easy to get an answer of puzzles or to decide the winner (loser) of games [2, 4, 10]. Such games and puzzles of interests include Nim, Hex, Peg Solitaire, Tetris, Geography, Amazons, Chess, Othello, Go, Poker, and so on. Recently, this field is sometimes called 'algorithmic combinatorial game theory' [2] to distinguish it from games arising from the other field, especially the classical economic game theory.

In this paper, we focus on one of the well-known and popular card games called UNO† and investigate it from the viewpoint of algorithmic combinatorial game theory to add it to the research list. More specifically, we propose mathematical models of UNO, which is one of the main purposes of this paper, and then examine their computational complexities. As a result, even a single-player version of UNO is computationally intractable, while we can show that the problem becomes rather easy under a certain restriction.

We organize this paper as follows: Section 2 introduces two mathematical models of UNO and their variants, and also defines UNO graphs. Among those models, Section 3 focuses on a single-player version of UNO, and investigates its complexities. In Section

† UNO® is a registered trademark of Mattel Corporation.

4, we argue with two-players' version of UNO, and show that it is PSPACE-complete. Finally, Section 5 concludes the paper.

2 Preliminaries

Games are often categorized from several aspects of properties that they have when we research it theoretically. Typical classifications are, for example, if it is multi-player or single-player, imperfect-information or perfect-information, cooperative or uncooperative, and so on [2, 8]. A single-player game is automatically perfect-information and cooperative, and is sometimes called a puzzle.

2.1 Game settings

UNO is one of the world-wide well-known and popular card games. It can be played by 2–10 players. Each player is dealt equal number of cards at the beginning of the game, where each (normal) card has its color and number (except for some special ones called 'action cards'). The basic rule is that each player plays in turn, and one can discard exactly one of his/her cards at hand in one's turn by matching the card with its color or number to the one discarded immediately before one. The objective of a single game is to be the first player to discard all the cards in one's hand before one's opponents. Thus, UNO is a (i) multi-player, (ii) imperfect-information, and (iii) uncooperative combinatorial game (see [3] for detailed rules of UNO).

In the real game setting of UNO, it is quite true that action cards play important roles to make this game complicated and interesting. However, in this paper, when we model the game mathematically, we concentrate on the most important aspect of the rules of UNO that a card has a color and a number and that one can discard a card only if its color or number match the card discarded immediately before one's turn. In addition to obeying this fundamental property, for theoretical simplicity, we set following assumptions on our mathematical models: (a) we do not take into account either action cards nor draw pile, (b) all the cards dealt to and at hand of any player are open during the game, i.e., perfect-information, (c) we do not necessarily assume that all the players have a same number of cards at the beginning of a game (unless otherwise stated), (d) any player acts rationally, e.g., any player is not allowed to skip one's turn intentionally, and (e) the first player can start a game by discarding any card he/she likes at hand.

2.2 Definitions and Notations

An UNO card has two attributes called *color* and *number*, and in general, we define a *card* to be a tuple $(x, y) \in X \times Y$, where $X = \{1, \ldots, c\}$ is a set of colors and $Y = \{1, \ldots, b\}$ is a set of numbers. Finite number of *players* $1, 2, \ldots, p$ (≥ 1) can join an UNO game. At the beginning of a single game of UNO, each card of a set of n cards C is dealt to one player among p players, i.e., each player i is initially given a set C_i of cards; $C_i = \{t_{i,1}, \ldots, t_{i,n_i}\}$ $(i = 1, \ldots, p)$. By definition, $\sum_{i=1}^{p} n_i = n$. Here, we assume that C is a multiple set, that is, there may be more than one card with the same color and the same number. We denote a card (x, y) dealt to player i by $(x, y)_i$. When the number of players is one, we omit the subscript without any confusion. Throughout the paper, we assume without loss of generality that player 1 is the first to play, and players $1, 2, \ldots, p$ play in turn in this order.

Player i can *discard* (or *play*) exactly one card currently at hand in his/her turn if the color or the number of the card is equal to each of the card discarded immediately before player i. In other words, we say that a card $t' = (x', y')_{i'}$ can be discarded immediately after a card $t = (x, y)_i$ if and only if $(x' = x \vee y' = y) \wedge i' = i + 1 \pmod{p}$. We also say that a card t' *matches* a card t when t' can be discarded after t. A discarded card is removed from a set of cards at hand of the player. A *discarding* (or *playing*) *sequence* (*of cards*) of a card set C is a sequence of cards $(t_{s_1}, \ldots, t_{s_k})$ such that $t_{s_i} \in C$ and $t_{s_i} \neq t_{s_j}$ $(i \neq j)$. A discarding sequence $(t_{s_1}, \ldots, t_{s_k})$ is *feasible* if $t_{s_{j+1}}$ matches t_{s_j} for $j = 1, \ldots, k - 1$.

In our mathematical models of UNO, we specify the problems by four parameters: number of players p, number of total cards n, number of colors c and the number of numbers b. Two values c and b are assumed to be unbounded unless otherwise stated.

2.3 Models

We now define two different versions of UNO, one is cooperative and the other is uncooperative.

> Uncooperative Uno
> Instance: the number of players p, and player i's card set C_i with c colors and b numbers.
> Question: determine the first player that cannot discard one's card any more.

We refer to this Uncooperative UNO with p players as Uncooperative Uno-p. This problem setting makes sense only if $p \geq 2$ since UNO played by a single player becomes automatically cooperative.

> Cooperative Uno
> Instance: the number of players p, player i's card set C_i with c colors and b numbers.
> Question: can all the players make player 1 win, i.e., make player 1's card set empty before any of the other players become finished.

We abbreviate Cooperative Uno played by p players as Cooperative Uno-p, or simply as Uno-p. This problem setting makes sense if the number of players p is greater than or equal to 1. In Uncooperative/Cooperative Uno, when the number of players is given by a constant, such as Uno-2, it implies that p is no longer a part of the input of the problem. In addition to the assumptions (a)–(e) on game settings described in Subsec. 2.1, we set one additional assumption which changes depending on whether the game is cooperative or uncooperative: any player that cannot discard any card at hand (f1) skips one's turn but still remains in the game and waits for the next turn in cooperative games, and (f2) is a loser in uncooperative games.

We define *UNO-p graph* as a directed graph to represent 'match' relationship between two cards in the entire card set. More precisely, a vertex corresponds to a card, and there is a directed arc from vertex u to v if and only if their corresponding cards

t_v matches (can be discarded immediately after) t_u. Let us consider UNO-1 graph, i.e., UNO-p graph in case that the number of players $p = 1$. In this case, a card t' matches t if and only if t matches t', that is, the 'match' relation is symmetric. This implies that UNO-1 graph becomes undirected. For UNO-2 graph, a card $t' = (x', y')_2$ matches $t = (x, y)_1$ if and only if t matches t', and therefore, UNO-2 graph also becomes undirected. Furthremore, since a player cannot play consecutively when the number of players $p \geq 2$, UNO-2 graph becomes bipartite. In general, since n cards of a card set C is dealt to p players at the beginning of a single UNO game, i.e., C is partitioned into $C_i = \{(x, y)_i\}$, UNO-p graph becomes a (restricted) p-partite graph whose partite sets correspond to C_i.

3 Cooperative UNO

In this section, we focus on the cooperative version of UNO, and discuss its complexity when the number of players is two or one.

3.1 Two-players' case

We first show that Uno-2 is intractable.

Theorem 1. Uno-2 *is NP-complete.*

Proof. Reduction from Hamiltonian Path (HP).

An instance of HP is given by an undirected graph G. The problem asks if there is a Hamiltonian path in G, and it is known to be NP-complete [7]. Here, we assume without loss of generality that G is connected and is not a tree, and hence that $|V(G)| \leq |E(G)|$. We transform an instance of HP into an instance of Uno-2 as follows. Let C_1 and C_2 be the card set of players 1 and 2, respectively. We define $C_1 = \{(i, i) \mid v_i \in V(G)\}$ and $C_2 = \{(i, j) \mid \{v_i, v_j\} \in E(G)\}$. Then, notice that the resulting UNO-2 graph G', which is bipartite, has partite sets X and Y ($X \cup Y = V(G')$) corresponding to $V(G)$ and $E(G)$, respectively, and represents vertex-edge incidence relationship of G (Fig. 1). Now we show that the answer of an instance of HP is yes if and only if the answer of an instance of Uno-2 is yes. If there is a Hamiltonian path, say $P = (v_{i_1}, v_{i_2}, \ldots, v_{i_n})$, in the instance graph of HP, then there is a feasible discarding sequence alternatively by player 1's and 2's as $((i_1, i_1)_1, (i_1, i_2)_2, (i_2, i_2)_1, \ldots, (i_{n-1}, i_{n-1})_1, (i_{n-1}, i_n)_2, (i_n, i_n)_1)$, which ends up player 1's card before player 2's. Conversely, if there is a feasible discarding sequence $((i_1, i_1)_1, (i_1, i_2)_2, (i_2, i_2)_1, \ldots, (i_{n-1}, i_{n-1})_1, (i_{n-1}, i_n)_2, (i_n, i_n)_1)$, it visits all the vertices in X of G' exactly once, and thus the corresponding sequence of vertices $(v_{i_1}, v_{i_2}, \ldots, v_{i_n})$ is a simple path visiting all the vertices in $V(G)$ exactly once, that is, a Hamiltonian path in G.

The size of an instance of Uno-2 is proportional to $|C_1| + |C_2|$. Since $|C_1| = |V(G)|$ and $|C_2| = |E(G)|$, the reduction is done in polynomial size in $|V(G)| + |E(G)|$, which is the input size of an instance of HP. This completes the proof. □

Corollary 1. Uno-2 *is NP-complete even when the number of cards of two players are equal.*

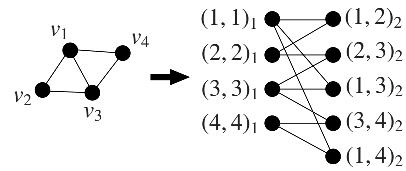

Fig. 1. Reduction from HP to Uno-2.

Proof. Reduction from Hamiltonian Path with specified starting vertex, which is known to be NP-complete [7].

We consider the same reduction in the proof of Theorem 1. As in that proof, we can assume $|C_1| \leq |C_2|$ without loss of generality. When $|C_1| = |C_2|$, we are done. If $|C_1| < |C_2|$, add $|C_2| - |C_1|$ cards $(n+2, n+2)$ and a single card $(n+2, n+1)$ to C_1, a single card $(i, n+1)$ $(i \in \{1, \ldots, n\})$ to C_2, and player 1 starts with card $(n+2, n+2)$. This forces the original graph G to specify a starting (or an ending) vertex of a Hamiltonian path to be v_i. □

3.2 Single-player's intractable case

In single-player's case, two different versions of UNO, cooperative and uncooperative ones, become equivalent. We redefine this setting as the following:

Uno-1 (Solitaire Uno)
Instance: a set C of n cards (x_i, y_i) $(i = 1, \ldots, n)$, where $x_i \in \{1, \ldots, b\}$ and $y_i \in \{1, \ldots, c\}$.
Question: determine if the player can discard all the cards.

Example 1. Let the card set C for player 1 is give by $C = \{(1,3), (2,2), (2,3), (2,3), (2,4), (3,2), (3,4), (4,1), (4,3)\}$. Then, a feasible discarding sequence using all the cards is $((1,3), (2,3), (2,4), (3,4), (3,2), (2,2), (2,3), (4,3), (4,1))$ in this order, for example, and the answer is yes. The corresponding UNO-1 graph is depicted in Fig. 2.

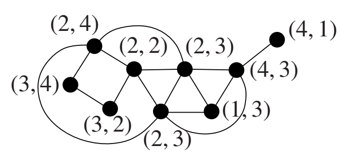

Fig. 2. An example of UNO-1 graph.

We here investigate some basic properties of UNO-1 graphs. In UNO-1 graphs, all the vertices whose corresponding cards have either the same color or number form a clique. A *line graph* $L(G)$ of a given graph G is a graph whose vertices are edges of G and $\{e, e'\} \in E(L(G))$ for $e, e' \in V(L(G))$ if and only if e and e' share endpoints in G. A graph that contains no induced $K_{1,3}$ is called *claw-free*, and line graphs are claw-free.

It is not so difficult to see that UNO-1 graphs are claw-free since at least two of the three cards that match a card must have the same color or number. Furthermore, we can observe the following fact.

Observation 1. A graph is UNO-1 if and only if it is a line graph of a bipartite graph.

Now we can easily understand that Uno-1 is essentially equivalent to finding a Hamiltonian path in UNO-1 graph. However, the following fact is known.

Theorem 2. [9] Hamiltonian Path *for line graphs of bipartite graphs is NP-complete.*

Therefore, as a corollary of this theorem, we unfortunately know that UNO is hard even for a single player.

Theorem 3. Uno-1 *is NP-complete.*

Here, we give a direct and concise proof of Theorem 3 for self-containedness and completeness instead of the one in [9], which further depends on [1].

Proof. A cubic graph is a graph each of whose vertex has degree 3. We reduce Hamiltonian Path for cubic graphs (HP-C), which is known to be NP-complete [6], to Uno-1.

Let an instance of HP-C be G. We transform G into a graph G', where

$$V(G') = \{(x, e) \mid x \in V(G), e = \{x, y\} \in E(G)\},$$
$$E(G') = \{((x, e), (y, e)) \mid e = \{x, y\} \in E(G)\} \cup \{((x, e_i), (x, e_j)) \mid e_i \neq e_j\}.$$

This transformation implies that any vertex $x \in V(G)$ is split into three new vertices (x, e_i) $(i = 1, 2, 3)$ to form a clique (triangle), while each incident edge e_i $(i = 1, 2, 3)$ to x becomes incident to a new vertex (x, e_i). (We call it a "node gadget" as shown in Fig. 3.) Then we prepare the card set C of the player of Uno-1 to be the set $V(G')$, where the color and the number of (x, e) are x and e, respectively. We can easily confirm that there is an edge $e = (t, t')$ in G' if and only if t and t' match, i.e., G' is the corresponding UNO-1 graph for card set C. Now it suffices to show that there is a Hamiltonian path in G of an instance of HP-C if and only if there is a Hamiltonian path in G'.

Suppose there is a Hamiltonian path, say $P = (v_{i_1}, \ldots, v_{i_n})$, in G. We construct a Hamiltonian path P' in G' from P as follows. Let $v_{i_{j-1}}, v_{i_j}, v_{i_{j+1}}$ be three consecutive vertices in P in this order, and let $e_1 = \{v_{i_{j-1}}, v_{i_j}\}$, $e_2 = \{v_{i_j}, v_{i_{j+1}}\}$ and $e_3 = \{v_{i_j}, v_{i_k}\}$ $(k \neq j-1, j+1)$. Then we replace these three vertices by the sequence of vertices $(v_{i_{j-1}}, e_1)$, (v_{i_j}, e_1), (v_{i_j}, e_3), (v_{i_j}, e_2), $(v_{i_{j+1}}, e_2)$ in G' to form a subpath in P'. For the starting two vertices v_{i_1} and v_{i_2}, we replace them by the sequence of vertices (v_{i_1}, e_1) $(e_1 \neq \{v_{i_1}, v_{i_2}\})$, (v_{i_1}, e_2) $(e_2 \neq \{v_{i_1}, v_{i_2}\})$, $(v_{i_1}, \{v_{i_1}, v_{i_2}\})$, $(v_{i_2}, \{v_{i_1}, v_{i_2}\})$ (same for the ending two vertices). We can now confirm that the resulting sequence of vertices P' in G' form a Hamiltonian path.

For the converse, we have to show that if there is a Hamiltonian path P' in UNO-1 graph G', then there is in G. If P' visits (v, e_i) $(i = 1, 2, 3)$ consecutively in any order (call it "consecutiveness") for any v (as shown in Fig. 4 (a1) or (a2)), then P' can be transformed into a Hamiltonian path P in G in an obvious way. Suppose not, that is, a Hamiltonian path P' in G' does not visit (v, e_i) $(i = 1, 2, 3)$ consecutively. It suffices to

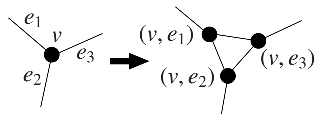

Fig. 3. A node gadget splits a vertex into three vertices to form a triangle.

show that such P' can be transformed into another path to satisfy the consecutiveness. There are two possible cases as shown in Fig. 4 (b') and (c'), both of which contain at least one end point of P' in (v, e_i). In case (b'), we can resolve this inconsecutiveness in (v, e_i) as shown in (b), which may result in case (c') in adjacent set of three vertices. In case (c'), in order to resolve it, we can transform it into (c), which does not contain inconsecutiveness any more.

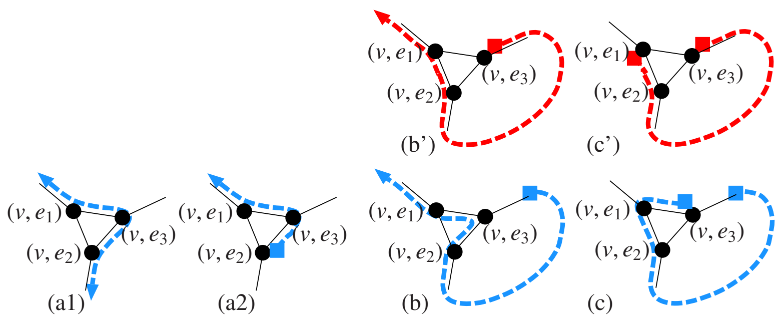

Fig. 4. Possible tours passing through a node gadget.

The reduction can be done in the size proportional to the size of an instance of HP-C. Thus, the proof is completed. □

3.3 Single-player's tractable case

In the remaining part of this section, we will show that such an intractable problem Uno-1 becomes tractable if the number of colors c is bounded by a constant. It will be solved by dynamic programming (DP) approach. To illustrate the DP for Uno-1, we first introduce a geometric view of UNO-1 graphs.

Since an UNO card (x, y) is an ordered pair of integer values standing for its color and number, it can be viewed as a (integer) lattice point in the 2-dimensional lattice plane. Then an UNO-1 graph is a set of points in that plane, where all the points with the same x- or y-coordinate form a clique. We call this way of interpretation a *geometric view* of UNO-1 graphs. The geometric view of an instance in Example 1 is shown in Fig. 5 (a). Now the problem Uno-1, which is equivalent to finding a Hamiltonian path in UNO-1 graphs, asks if, for a given set of points in the plane and starting and ending at appropriate different points, one can visit all the points exactly once under the condition that only axis-parallel moves are allowed at each point (Fig. 5 (b)).

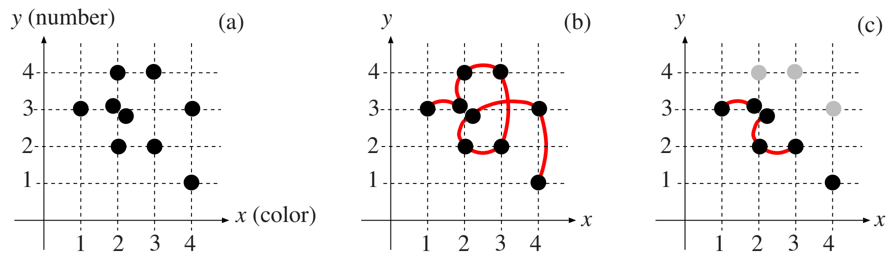

Fig. 5. (a) Geometric view of a UNO-1 graph, where all the edges are omitted, (b) a Hamiltonian path in the UNO-1 graph, and (c) a set of subpaths in the subgraph of the UNO-1 graph induced by the first 6 points; it shows $h_{\{1,2\}} = 1$, $v_{(2,3)} = 1$ and $d_{\{4,4\}} = 1$.

Strategy. Let C be a set of n points and G be an UNO-1 graph defined by C. Then a subgraph P forms a Hamiltonian path if and only if it is a single path that spans G. Suppose a subgraph P is a spanning path of G. If we consider a subset C' of the point set C, then $P[C']$ (the subgraph of P induced by C') is a set of subpaths that spans $G[C']$ (Fig. 5 (c)). We count and maintain the number of sets of subpaths by classifying subpaths into three disjoint subsets according to the types of their two endpoints.

Starting with the empty set of points, the DP proceeds by adding a new point according to a fixed order by updating the number of sets of subpaths iteratively. Finally when the set of points grows to C, we can confirm the existence of a Hamiltonian path in G by checking the number of sets of subpaths consisting of a single subpath (without isolated vertices). Remark that, throughout this DP, we regard for convenience that an isolated vertex by itself contains a (virtual) path starting and ending at itself that spans it.

Mechanism. To specify a point to be added in an iteration of the DP, we define a relation $\prec$ on the point set C, where $x(t)$ and $y(t)$ are x- and y-coordinates of a point t, respectively: Let t and t' be two points in C, then $t \prec t' \iff y(t) < y(t')$ or $(y(t) = y(t') \wedge x(t) < x(t'))$. When $t = t'$, a tie breaks arbitrary. This relation $\prec$ defines a total order on C, and we refer n points in C to $t_1, \ldots, t_n$ according to the increasing order of $\prec$. We also define $C_\ell = \{t_i \mid 1 \leq i \leq \ell\}$. Now points are added from t_1 to t_n, and consider when a new point $t_\ell = (x(t_\ell), y(t_\ell))$ is added to $C_{\ell-1}$. It must be added either to two, one or zero endpoints of different subpaths to form a new set of subpaths.

Now let $\mathcal{P}(\ell)$ be a family of sets of subpaths spanning $G[C_\ell]$. (Recall that we regard that an isolated vertex contains a path spanning itself.) Then we classify subpaths in a set of subpaths $\mathcal{P} \in \mathcal{P}(\ell)$ in the following manner: for any subpath $P \in \mathcal{P}$ and the y-coordinates of its two endpoints, either (i) both equal $y(t_\ell)$ (type-h), (ii) exactly one of two equals $y(t_\ell)$ (type-v), or (iii) none equals $y(t_\ell)$ (type-d) holds. We count the number of such three types of subpaths in $\mathcal{P}$ further by classifying them by the x-coordinates of their endpoints. (Notice that types-h, -d are symmetric but type-v is not with respect to x-coordinate.) For this purpose, we prepare some subscript sets: a set of subscripts $K = \{1, \ldots, c\}$, sets of unordered pair of subscripts $I = \binom{K}{2}$ and $I^+ = I \cup \{\{i, i\} \mid i \in K\}$, and sets of ordered pair of subscripts $J = K \times K$ and $J^- = J - \{(i, i) \mid i \in K\}$.

We now introduce the following parameters h, v and d to count the number of subpaths in $\mathcal{P}$ ($\in \mathcal{P}(\ell)$) (see Fig. 5 (c)):

$h_{\{i,i'\}}$: #subpaths in $\mathcal{P}$ with endpoints $(x_i, y(t_\ell))$ and $(x_{i'}, y(t_\ell))$ for $\{i, i'\} \in I^+$,
$v_{(i,i')}$: #subpaths in $\mathcal{P}$ with endpoints $(x_i, y(t_\ell))$ and $(x_{i'}, y')$ for $(i, i') \in J$ and $y' < y(t_\ell)$,
$d_{\{i,i'\}}$: #subpaths in $\mathcal{P}$ with endpoints (x_i, y') and $(x_{i'}, y'')$ for $\{i, i'\} \in I^+$ and $y', y'' < y(t_\ell)$.

Then we define a $(2|I^+| + |J|)$-dimensional vector $z(\mathcal{P})$ for a set of subpaths $\mathcal{P}$ ($\in \mathcal{P}(\ell)$) as $z(\mathcal{P}) = (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d}) = (\langle h_{\{1,1\}}, \ldots, h_{\{1,c\}}, h_{\{2,2\}}, \ldots, h_{\{2,c\}}, h_{\{3,3\}}, \ldots, h_{\{c,c\}}\rangle; \langle v_{\{1,1\}}, \ldots, v_{\{1,c\}}, v_{\{2,1\}}, v_{\{2,2\}}, \ldots, v_{\{2,c\}}, v_{\{3,1\}}, \ldots, v_{\{c,c\}}\rangle; \langle d_{\{1,1\}}, \ldots, d_{\{1,c\}}, d_{\{2,2\}}, \ldots, d_{\{2,c\}}, d_{\{3,3\}}, \ldots, d_{\{c,c\}}\rangle)$. Finally, for a given vector $(\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})$, we define the number of sets $\mathcal{P}$ satisfying $z(\mathcal{P}) = (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})$ in a family $\mathcal{P}(\ell)$ by $f(\ell, (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d}))$, i.e., $f(\ell, (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})) = \left|\{\mathcal{P} \mid \mathcal{P} \in \mathcal{P}(\ell), z(\mathcal{P}) = (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})\}\right|$. Now the objective of the DP is to determine if there exists a vector $(\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})$ such that $f(n, (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d})) \geq 1$, where all the elements in $\boldsymbol{h}$, $\boldsymbol{v}$ and $\boldsymbol{d}$ are 0 except for exactly one element is 1.

Recursion. As we explained, the DP proceeds by adding a new point t_ℓ to $C_{\ell-1}$. When t_ℓ is added, it is connected to either 0, 1 or 2 endpoints of existing different paths, where each endpoint has $y(t_\ell)$ or $x(t_\ell)$ in its coordinate. The recursion of the DP is described just by summing up all possible combinations of these patterns. We treat it by dividing them into three cases, one of which has two subcases: (a) a set of base cases; (b) a case in which t_ℓ is added as the first point whose y-coordinate is $y(t_\ell)$, and (b1) as an isolated vertex, or (b2) as to be connected to an existing path; (c) all the other cases.

Now we can give the DP formula for computing $f(\ell; (\boldsymbol{h}; \boldsymbol{v}; \boldsymbol{d}))$, however, we just explain the idea of the DP in Fig. 6 by illustrating one of the cases appearing in the DP (see [3] for full description of this recursion). In this example, consider a subpath in a graph induced by C_ℓ whose two endpoints have $x_{i'}$ and x_j in their x-coordinates. It will be counted in $h_{\{i',j\}}$. Then this subpath can be generated by adding point t_ℓ to connect to two paths in a graph induced by $C_{\ell-1}$, the one whose one endpoint is $(x_i, y(t_\ell))$ (counted in $v_{(i,i')}$), and the other whose one endpoint is (k, y) ($y < y(t_\ell)$) (counted in $d_{\{j,k\}}$). The number of such paths is the sum of those for all the combinations of i, i' and j.

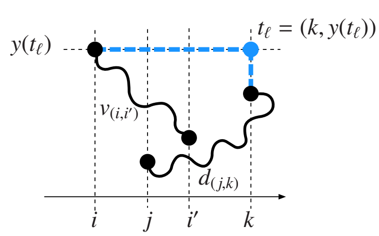

Fig. 6. An example case of the DP.

Timing analysis. We first count possible combinations of arguments for f. Since ℓ varies from 0 to n, there are $\Theta(n)$ possible values. All of $\boldsymbol{h}$, $\boldsymbol{v}$ and $\boldsymbol{d}$ have $\Theta(c^2)$ elements, each of which can have $O(n)$ possible values, and therefore $O(n^{c^2})$ possible values in all. To compute a single value of f, it requires $O(n^4)$ lookups of previously computed values of f in case (c), while $O(n^{3c^2}) \times O(n^2)$ lookups and check-sums in cases (b1) and (b2), which is greater than $O(n^4)$. Therefore, the total running time for this DP is $\Theta(n) \times O(n^{3c^2}) \times O(n^{3c^2+2}) = O(n^{6c^2+3}) = n^{O(c^2)}$, which is polynomial in n when c is a constant.

Since the role of colors and numbers are symmetric in UNO games, we have the following results.

Theorem 4. Uno-1 *is in P if b (the number of numbers) or c (the number of colors) is a constant.*

4 Uncooperative UNO

In this section, we deal with the uncooperative version of UNO. Especially, we show that it is intractable even for two player's case. For this purpose, we consider the following version of Generalized Geography, which is played by two players.

> Generalized Geography
> Instance: a directed graph, and a token placed on an initial vertex.
> Question: a turn is to move the token to an adjacent vertex, and then to remove the vertex moved from from the graph. Player 1 and 2 take turns, and the first player unable to move loses. Determine the loser.

It is well-known that Generalized Geography is PSPACE-complete [10], and a stronger result is presented.

Theorem 5. [10] Generalized Geography *for bipartite graphs is PSPACE-complete.*

Now we show the hardness result for Uncooperative Uno-2.

Theorem 6. Uncooperative Uno-2 *is PSPACE-complete.*

Proof. Reduction from Generalized Geography for bipartite graphs (GG-B).

Let (directed) bipartite graph G with $V(G) = X \cup Y$ be an instance of GG-B, where X and Y are two partite sets, and let r $(\in X)$ be an initial vertex. To construct a corresponding Uncooperative Uno-2 instance, we first transform G into another graph G' where

$$V(G') = \{u_s, u_t, u_c \mid u \in V(G)\},$$
$$E(G') = \{(u_t, u_c), (u_c, u_s) \mid u \in V(G)\} \cup \{(u_s, v_t) \mid (u, v) \in E(G)\}$$

(Fig. 7). By construction, we can confirm that G' is a bipartite graph with $V(G') = X' \cup Y'$, where $X' = \{u_s, u_t \mid u \in X\} \cup \{u_c \mid u \in Y\}$ and $Y' = \{u_s, u_t \mid u \in Y\} \cup \{u_c \mid u \in X\}$. We let $r' = r_t$ $(\in X')$ be an initial vertex. It is easy to confirm that player 1 can win the game GG-B on G if and only if the player wins on G'. Then we prepare card sets C_i for players i $(= 1, 2)$ by

$$\begin{aligned} C_1 = \{&(x, e), (e, y) \mid e = (x, y) \in E(G'), x \in X', y \in Y'\} \\ &\cup \{(e, e) \mid e = (y, x) \in E(G'), x \in X', y \in Y'\}, \\ C_2 = \{&(y, e), (e, x) \mid e = (y, x) \in E(G'), x \in X', y \in Y'\} \\ &\cup \{(e, e) \mid e = (x, y) \in E(G'), x \in X', y \in Y'\}. \end{aligned}$$

This means that we prepare three cards for each arc e in $E(G')$, one for player i and two for player $3 - i$ (Fig. 8).

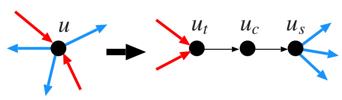

Fig. 7. Split a vertex into two edges so that edges correspond to cards.

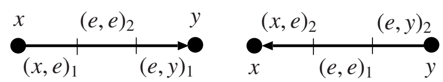

Fig. 8. Prepare three cards $(x,e)_1$, $(e,e)_2$ and $(e,y)_1$ for an arc $e = (x,y)$, and three cards $(e,y)_2$, $(e,e)_1$ and $(x,e)_2$ for an arc $e = (y,x)$.

Now we show that player 1 can win in an Uncooperative Uno-2 instance if and only if player 1 can win in an GG-B instance G' and s'. To show this, it suffices to show that any feasible playing sequence by players 1 and 2 in an GG-B instance corresponds to a feasible discarding sequence alternatively by players 1 and 2 in the corresponding Uncooperative Uno-2 instance, and vice versa.

Suppose a situation that player 2 has just discarded a card. The discarded card belongs to either one of the following five cases: (i) (e,x) for $e = (y,x)$, (ii) (y,e) for $e = (y,x)$, (iii) (e,e) for $e = (x,y)$. Among those, for cases (ii) and (iii), since player 1 starts the game (player 1 always played before player 2's turn), there exists exactly one card (outgoing arc) that matches the one discarded by player 2 from the end vertex of the arc corresponding to the card. This forces to traverse G' along the directed arc (in forward direction), which implies to remove corresponding end vertex from G'. The only case we have to care about is case (i), where there may be multiple choices for player 1. In this case, once player 1 discarded one of match cards, the player will never play another match card afterwards, since the only card that can be discarded immediately before it has played and used up. This implies that vertex x is removed from G'. (The argument is symmetric for player 1 except that the initial card is specified.)

Now we verify that Uncooperative Uno-2 is in PSPACE. For this, consider a search tree for Uncooperative Uno-2, whose root is for player 1 and every node has outgoing arcs corresponding to each player's possible choices. Since the number of total cards for the two players is n, the number of choices at any turn is $O(n)$ and since at least one card is removed from either of the player's card set, the number of depth of the search tree is bounded by $O(n)$. Therefore, it requires polynomial space with respect to the input size. Thus the proof is completed. □

5 Concluding Remarks

In this paper, we focused on UNO, the well-known card game, and gave two mathematical models for it; one is cooperative (to make a specified player win), and the other is uncooperative (to decide the player not to be able to play). As a result of analyzing their complexities, we showed that these problems are difficult in many cases, however, we also showed that a single-player's version is solvable in polynomial time under a certain restriction.

As for an obvious future work, we can try gaining speedup in dynamic programming for Uno-1 with constant number of colors by better utilizing its geometric properties. In this direction, it may be quite natural to ask if Uno-1 is fixed-parameter tractable. Another probable direction is to investigate UNO-1 graphs from the structural point of view, since they form a subclass of claw-free graphs and seem to have interesting properties by themselves. It is also quite probable to modify our models more realistic, e.g., to take draw pile into account (as an additional player), to make all players' cards not open, and so on.

Based on our mathematical models, it is not so difficult to invent several variations or generalizations of UNO games, even for Uno-1 (single-player's version). Among them, we can generalize an UNO card from 2-tuple (2-dimensional) to *d*-tuple, that is, D-dimensional Uno-1 with appropriate modifications to 'match' relation of cards. Another one is Minimum Card Fill-in, that is, given a no instance for Uno-1, find a minimum number of cards to be added to make it to be a yes instance.

Acknowledgments We deeply appreciate Nicholas J. A. Harvey at University of Waterloo, Canada, for fruitful discussions with his deep insight at the early stage of this manuscript. We also thank for the anonymous referees for their valuable comments.

References

1. A. A. Bertossi. The edge Hamiltonian path problem is NP-complete. *Information Processing Letters* 13, 157–159 (1981).
2. E. D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. *Lecture Notes in Computer Science*, Vol. 2136, 18–32, Springer (2001).
3. E. D. Demaine, M. L. Demaine, R. Uehara, T. Uno and Y. Uno. The complexity of UNO. CoRR abs/1003.2851 (2010).
4. S. Even and R. E. Tarjan. A combinatorial problem which is complete in polynomial space. *J. ACM* 23, 710–719 (1976).
5. M. Gardner. *Mathematical Games: The Entire Collection of his Scientific American Columns*. The Mathematical Association of America (2005).
6. M. R. Garey, D. S. Johnson and R. E. Tarjan. The planar Hamiltonian circuit is NP-complete. *SIAM Journal of Computing* 5, 704–714 (1976).
7. M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company (1979).
8. R. A. Hearn and E. D. Demaine. *Games, Puzzles, and Computation*. A. K. Peters (2009).
9. T.-H. Lai and S.-S. Wei. The edge Hamiltonian path problem is NP-complete for bipartite graphs. *Information Processing Letters* 46, 21–26 (1993).
10. D. Lichtenstein and M. Sipser. GO is polynomial-space hard. *J. ACM* 27, 393–401 (1980).